

COMPATIBLE GAUGE APPROACHES FOR $H(\text{div})$ EQUATIONS

P.B. BOCHEV*, C.M. SIEFERT*, R.S. TUMINARO*, J. XU†, AND Y. ZHU†

Abstract. We are concerned with the compatible gauge reformulation for $H(\text{div})$ equations and the design of fast solvers of the resulting linear algebraic systems as in [5]. We propose an algebraic reformulation of the discrete $H(\text{div})$ equations along with an algebraic multigrid (AMG) technique for the reformulated problem. The reformulation uses discrete Hodge decompositions on co-chains to replace the discrete $H(\text{div})$ equations by an equivalent 2×2 block linear system whose diagonal blocks are discrete Hodge Laplace operators acting on 2-cochains and 1-cochains respectively. We illustrate the new technique, using the lowest order Raviart-Thomas elements on structured tetrahedral mesh in three dimension and present computational results.

1. Introduction. In this paper, we consider general second order elliptic operators over the Lipschitz polyhedral domain Ω in 3D. Specifically, let Ω be a bounded, simply connected, and contractible domain in \mathbb{R}^3 with Lipschitz boundary $\partial\Omega$. We are looking at the compatible discretization of the following model equation:

$$\begin{cases} -\nabla(\lambda \nabla \cdot u) + \frac{1}{\mu} u &= f & \text{in } \Omega, \\ \frac{1}{\mu} u \cdot n &= 0 & \text{on } \Gamma, \\ \lambda \nabla \cdot u &= 0 & \text{on } \Gamma^*, \end{cases} \quad (1.1)$$

where $\partial\Omega = \Gamma \cup \Gamma^*$ and $\Gamma \cap \Gamma^* = \emptyset$. Here, we assume that λ and μ are positive throughout the domain, but may possibly vary widely.

The variational formulation of problem (1.1) leads naturally to the Hilbert space $H(\text{div})$ given by

$$H(\text{div}) := \left\{ u \in (L^2(\Omega))^3 \mid \nabla \cdot u \in L^2(\Omega) \right\}.$$

This equation is ubiquitous in problems arising in fluid and solid mechanics [6, 10]. It occurs, in particular, in the solution of second order elliptic partial differential equations (PDE) by first order least-squares methods or by mixed methods with augmented Lagrangians, see [1, 11, 18, 19] and the references cited therein. The importance of $H(\text{div})$ -related problems has prompted vigorous research into efficient multilevel schemes, see [1, 11, 12, 18, 19].

The method to be developed in the current paper follows closely the idea of the recent work of Bochev, Hu, Siefert and Tuminaro [5] for Maxwell's equations. Specifically, we propose an algebraic reformulation of the discrete $H(\text{div})$ equations along with a new AMG technique for this reformulated problem. The reformulation process take advantage of a discrete Hodge decomposition on co-chains to replace the discrete $H(\text{div})$ equations by an equivalent 2×2 block linear system whose diagonal blocks are discrete Hodge Laplace operators acting on 2-cochains and 1-cochains, respectively. The new AMG algorithm in this paper makes use of the Hiptmair smoother ([11]) on the fine mesh, uses the canonical interpolations Π_h^{div} and Π_h^{curl} on $H(\text{div})$ and $H(\text{curl})$ to construct the grid-transfer operators, and then uses the standard AMG methods for Laplace-type problems on the coarse meshes.

The rest of the paper is organized as follows. Section 2 reviews basic facts about the discretization framework used in the paper. In Section 3, we apply this framework

*Sandia National Laboratories, Computational Math/Algorithms, {pbboche, csiefer, rstumin}@sandia.gov

†Penn. State University for Department of Mathematics, {xu, zhu.y}@math.psu.edu

to obtain a compatible discretization for the $H(\text{div})$ equations and its equivalent reformulation. AMG solvers for the reformulated system are developed in Section 4. In Section 5 we present computational results in three dimension that illustrate the new technique in the context of smoothed aggregation AMG. In all experiments we use finite element discretizations based on the lowest order Raviart-Thomas element and lowest order Nédélec element on structured tetrahedral elements.

2. Compatible discretization framework. In this section, we give a short introduction of a general framework for compatible discretizations developed in [3]. This framework is based on algebraic topology and includes certain finite element [4, 17], finite volume [15], and finite difference [16] schemes as particular cases. As a result, the AMG algorithm developed in this paper is readily applicable to discrete problems generated by any of these schemes. The presentation here is almost the same as [5, Section3]. We include this section just for the sake of completeness.

2.1. Computational grid. We consider computational grids Ω^h consisting of 0-cells (nodes), 1-cells (edges), 2-cells (faces), and 3-cells (volumes). Formal linear combinations of k -cells are called k -chains [8]. The sets of k -chains forming Ω^h are denoted by C_k . We will assume that Ω^h is such that the collection $\{C_0, C_1, C_2, C_3\}$ is a complex, i.e., for any $c \in C_k$, $\partial_k c \in C_{k-1}$, where $\partial_k : C_k \mapsto C_{k-1}$ is the boundary operator on k -chains [7]. Together with the identity $\partial_k \partial_{k+1} = 0$ this gives rise to the exact sequence

$$0 \longleftarrow C_0 \xleftarrow{\partial_1} C_1 \xleftarrow{\partial_2} C_2 \xleftarrow{\partial_3} C_3 \longleftarrow 0. \quad (2.1)$$

The dual of C_k is denoted by C^k and its members are called k -cochains [8]. While C_k and C^k are isomorphic, they have different meanings in our discretization framework. The sets C_k represent the physical objects that form the grid, while C^k are collections of real numbers associated with the grid objects. For example, $c_1 \in C_1$ is a formal sum of (oriented) grid edges, while its isomorphic image $c^1 \in C^1$ is a set of real numbers¹ assigned to the edges of c_1 .

Therefore, the elements of C^0 provide values associated with the 0-cells (grid nodes); the elements of C^1 are values associated with the 1-cells (grid edges); C^2 contains values assigned to the 2-cells (grid faces) of the grid, and C^3 are the values assigned to the 3-cells (grid volumes). We will use C^0 and C^3 to approximate scalar functions and C^1 and C^2 - to approximate vector functions.

The symbols C_Γ^k will denote the subspaces of C^k constrained by zero on the Dirichlet boundary Γ for $k = 0, 1, 2$. Such spaces are needed to approximate scalar and vector functions subject to appropriate boundary conditions².

2.2. Natural operators. Let $\langle \cdot, \cdot \rangle$ denote the duality pairing of C_k and C^k . The adjoint of ∂_k , defined by $\langle a, \partial_k c \rangle = \langle \delta_k a, c \rangle$, induces an operator $\delta_k : C_\Gamma^k \mapsto C_\Gamma^{k+1}$ called the coboundary. This operator satisfies $\delta_{k+1} \delta_k = 0$ and gives rise to the exact sequence

$$\mathbb{R} \longrightarrow C_\Gamma^0 \xrightarrow{\delta_0} C_\Gamma^1 \xrightarrow{\delta_1} C_\Gamma^2 \xrightarrow{\delta_2} C^3 \longrightarrow 0. \quad (2.2)$$

¹Clearly, C^k are isomorphic to $\mathbf{R}^{\tilde{k}}$, where $\tilde{k} = \dim C^k$. For simplicity, the isomorphic image of the cochain $c^k \in C^k$ in $\mathbf{R}^{\tilde{k}}$ will be denoted by the same symbol.

²For example, C_Γ^0 approximates scalar functions such that $\phi = 0$ on Γ ; C_Γ^1 can be used to approximate vector fields \mathbf{E} such that $\mathbf{n} \times \mathbf{E} = 0$ on Γ . The space C_Γ^2 is appropriate for vector fields \mathbf{B} that have a vanishing normal component on Γ .

It is not hard to see that the matrix representation \mathbb{D}_k of δ_k is the signed incidence matrix between C^k and C^{k+1} . Following [14] we call \mathbb{D}_0 , \mathbb{D}_1 , and \mathbb{D}_2 *natural* approximations of the gradient, curl and divergence operators. Note that from $\delta_{k+1}\delta_k = 0$ it follows that

$$\mathbb{D}_{k+1}\mathbb{D}_k = 0; \quad k = 0, 1, 2, \quad (2.3)$$

and so our natural operators mimic the well-known vector calculus identities $\nabla \times \nabla = 0$, and $\nabla \cdot \nabla \times = 0$. In [13], it is pointed out that natural operations are not enough to provide compatible discretizations of the basic second order operators because their ranges and domains do not match. For example, we cannot approximate $\nabla \times \nabla \times$ by $\mathbb{D}_1\mathbb{D}_1$ because \mathbb{D}_1 is in general a rectangular matrix. The number of its columns and rows equals the number of 1-cells and 2-cells in the grid, which are not the same.

2.3. Metric structures and derived operators. Let $\mathbb{M}_k : C_\Gamma^k \mapsto C_\Gamma^k$; $k = 0, 1, 2, 3$ denote symmetric positive definite matrices. The matrix \mathbb{M}_k endows C_Γ^k with an inner product structure,

$$(a^k, b^k)_{C^k} = (a^k)^T \mathbb{M}_k (b^k). \quad (2.4)$$

The matrices \mathbb{M}_0 and \mathbb{M}_3 approximate weighted L^2 inner products of scalar functions:

$$\mathbb{M}_0 \longrightarrow \int_\Omega \gamma p \hat{p} d\Omega; \quad \mathbb{M}_3 \longrightarrow \int_\Omega \lambda \phi \hat{\phi} d\Omega,$$

while \mathbb{M}_1 and \mathbb{M}_2 approximate the weighted L^2 inner products of vector functions

$$\mathbb{M}_1 \longrightarrow \int_\Omega \sigma \mathbf{E} \hat{\mathbf{E}} d\Omega; \quad \mathbb{M}_2 \longrightarrow \int_\Omega \mu^{-1} \mathbf{B} \hat{\mathbf{B}} d\Omega.$$

We will also use the notation $\mathbb{M}_0(\gamma)$, $\mathbb{M}_1(\sigma)$, $\mathbb{M}_2(\mu^{-1})$ and $\mathbb{M}_3(\lambda)$ to show the dependency of the coefficients of these mass matrices explicitly.

We define the derived operator $\mathbb{D}_k^* : C_\Gamma^{k+1} \mapsto C_\Gamma^k$ as the adjoint of \mathbb{D}_k with respect to the inner product (2.4):

$$(\mathbb{D}_k^* a^{k+1}, b^k)_{C^k} = (a^{k+1}, \mathbb{D}_k b^k)_{C^{k+1}}. \quad (2.5)$$

From (2.5) it is easy to see that for $k = 0, 1, 2$

$$\mathbb{D}_k^* = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1}. \quad (2.6)$$

The matrices \mathbb{D}_2^* , \mathbb{D}_1^* and \mathbb{D}_0^* provide a second set of discrete differential operators. Specifically, they are approximations of scaled gradient, curl and divergence operators

$$\mathbb{D}_2^* \rightarrow -\mu \nabla \lambda; \quad \mathbb{D}_1^* \rightarrow \sigma^{-1} \nabla \times \mu^{-1}; \quad \mathbb{D}_0^* \rightarrow -\gamma^{-1} \nabla \cdot \sigma,$$

augmented with the boundary conditions

$$\lambda \phi = 0; \quad \mathbf{n} \times \mu^{-1} \mathbf{B} = 0; \quad \text{and} \quad \mathbf{n} \cdot \sigma \mathbf{E} = 0 \quad \text{on } \Gamma^*,$$

respectively. Using (2.6) and (2.3)

$$\mathbb{D}_k^* \mathbb{D}_{k+1}^* = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1} \mathbb{M}_{k+1}^{-1} \mathbb{D}_{k+1}^T \mathbb{M}_{k+2} = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{D}_{k+1}^T \mathbb{M}_{k+2} = 0,$$

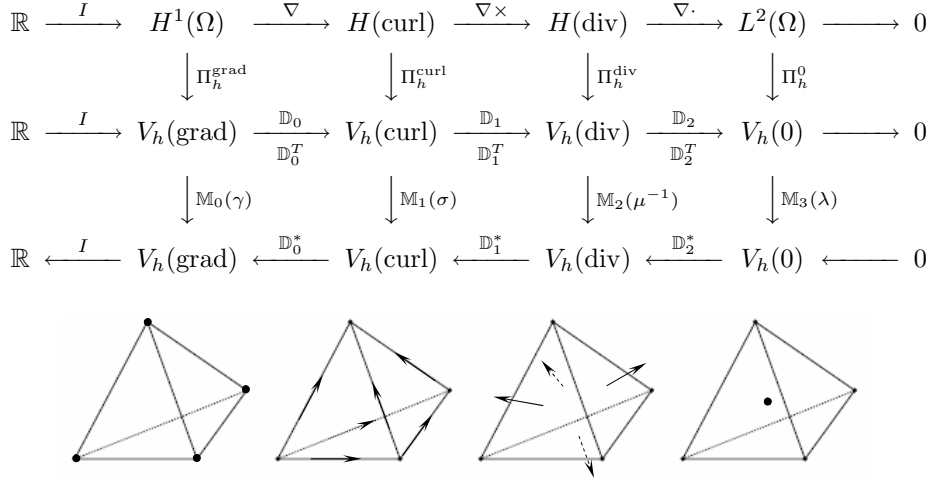


FIG. 2.1. De Rham Complex and the lowest order finite element spaces

and so, the basic vector calculus identities hold for the derived operators as well. The commuting diagram, and the relationships among the operators defined above can be summarized in Figure 2.1. Here, the operators Π_h^{grad} , Π_h^{curl} , Π_h^{div} , and Π_h^0 are the canonical interpolations on $H^1(\Omega)$, $H(\text{curl})$, $H(\text{div})$, and $L^2(\Omega)$ to the corresponding finite element spaces $V_h(\text{grad})$, $V_h(\text{curl})$, $V_h(\text{div})$, and $V_h(0)$ respectively. The lower half of the commuting diagram above presents the relationships among the operators. For example, from this diagram we can easily find out that

$$\mathbb{D}_2^* = \mathbb{M}_2^{-1} \mathbb{D}_2^T \mathbb{M}_3.$$

Because the range of \mathbb{D}_k is contained in the domain of \mathbb{D}_k^* and vice versa we can use the natural and the derived operators to define discrete versions of the basic second order differential operators, including a discrete Hodge Laplace operator. Specifically, for $k = 0, 1, 2$ we have the second order operators

$$\mathbb{D}_k^* \mathbb{D}_k = \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1} \mathbb{D}_k : C_\Gamma^k \mapsto C_\Gamma^k \quad (2.7)$$

$$\mathbb{D}_k \mathbb{D}_k^* = \mathbb{D}_k \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1} : C_\Gamma^{k+1} \mapsto C_\Gamma^{k+1} \quad (2.8)$$

and the discrete Hodge Laplacian

$$\mathbb{L}_k : C_\Gamma^k \mapsto C_\Gamma^k ; \quad \mathbb{L}_k = \mathbb{D}_k^* \mathbb{D}_k + \mathbb{D}_{k-1} \mathbb{D}_{k-1}^* ; \quad k = 0, 1, 2, 3 \quad (2.9)$$

with the understanding that $\mathbb{D}_3 = 0$ and $\mathbb{D}_{-1} = 0$.

The discrete operators in (2.7)-(2.9) approximate basic second order elliptic differential operators. In §3.1 we will use these operators to motivate and explain our reformulation strategy.

Similar to [5], we also introduce a second set of inner products defined by the matrices $\tilde{\mathbb{M}}_k$, ($k = 0, 1, 2, 3$) that uses a unit weight, i.e.,

$$\tilde{\mathbb{M}}_k \rightarrow \int_\Omega u^k v^k d\Omega, \quad u^k, v^k \in C_\Gamma^k.$$

These inner products can be used to define a second set of derived operators $\tilde{\mathbb{D}}_k^* : C_\Gamma^{k+1} \mapsto \tilde{\mathbb{D}}_k^*$ given

$$\tilde{\mathbb{D}}_k^* = \mathbb{M}_k^{-1} \mathbb{D}_k^T \tilde{\mathbb{M}}_{k+1}, k = 0, 1, 2$$

respectively, and such that $\tilde{\mathbb{D}}_k^* \tilde{\mathbb{D}}_{k+1}^* = 0$. These operators give rise to the discrete Hodge Laplace operators

$$\tilde{\mathbb{L}}_k : C_\Gamma^k \mapsto C_\Gamma^k; \quad \tilde{\mathbb{L}}_k = \tilde{\mathbb{D}}_k^* \mathbb{D}_k;$$

that are different versions of \mathbb{L}_k respectively.

The following general result from [3] provides the results needed for the reformulation of the discrete $H(\text{div})$ equation.

THEOREM 2.1. *The size of the kernel of the analytic and discrete Hodge Laplacians is the same.*

Theorem 2.1 reveals that the null-space of the discrete Hodge Laplacian and, by extension the structure of the discrete Hodge decomposition of discrete functions in C_Γ^k , are topological invariants that are independent of the particular choice of metric, i.e., the matrices \mathbb{M}_k . As a result, the assertion of this theorem is valid for both $\mathbb{L}_0, \mathbb{L}_1, \mathbb{L}_2$, and $\tilde{\mathbb{L}}_0, \tilde{\mathbb{L}}_1, \tilde{\mathbb{L}}_2$. The properties of these operators, relevant to the reformulation process, are summarized in the following corollary, which is a generalization of [5, Corollary 3.2].

COROLLARY 2.2. *Assume that Ω is contractible. Then, every $u^k \in C_\Gamma^k$ ($k = 1, 2$) has the discrete Hodge decomposition*

$$u^k = \mathbb{D}_{k-1} p^{k-1} + \tilde{\mathbb{D}}_k^* b^{k+1} \quad (2.10)$$

where $p^{k-1} \in C_\Gamma^{k-1}$ and $b^{k+1} \in C_\Gamma^{k+1}$ solve the equations

$$\tilde{\mathbb{D}}_{k-1}^* \mathbb{D}_{k-1} p^{k-1} = \tilde{\mathbb{D}}_{k-1}^* u^k \quad \text{and} \quad \mathbb{D}_k \tilde{\mathbb{D}}_k^* b^{k+1} = \mathbb{D}_k u^k, \quad (2.11)$$

respectively.

3. Compatible discretization of $H(\text{div})$ equation. Using the discrete operators defined in the last section, a compatible discretization of the $H(\text{div})$ equation (1.1) is straightforward. Specifically, we approximate u by a 2-cochain $u^2 \in C_\Gamma^2$ that is associated with the 2-cells (the faces) of the mesh that are not in Γ . Then the compatible discrete version of the $\nabla \nabla \cdot$ operator is provided by the second order discrete operator $\mathbb{D}_2^* \mathbb{D}_2$. As a result, the compatible, fully discrete equation of (1.1) is given by

$$(\mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \mathbb{M}_2) u^2 = f^2, \quad (3.1)$$

with the matrix \mathbb{M}_3 containing the material parameter λ and the matrix \mathbb{M}_2 containing μ^{-1} and $f^2 \in C_\Gamma^2$ is a discrete version of f in (1.1). An equivalent “weak” form of (3.1) is given by the variational equation: seek $u^2 \in C_\Gamma^2$ such that

$$(u^2, \hat{u}^2)_{C^2} + (\mathbb{D}_2 u^2, \mathbb{D}_2 \hat{u}^2)_{C^3} = (f^2, \hat{u}^2) \quad \forall \hat{u}^2 \in C_\Gamma^2. \quad (3.2)$$

3.1. Reformulation. Following [5] for Maxwell's equations, we intend on forming the Hodge Laplacian, which here corresponds to adding a $\nabla \times \nabla \times$ term, namely

$$\mathbb{L}_2 = \mathbb{D}_2^* \mathbb{D}_2 + \mathbb{D}_1 \mathbb{D}_1^*. \quad (3.3)$$

The following main theorem states an analogue of Theorem 4.2 in [5].

THEOREM 3.1. *Assume that u^2 is a solution of (3.1) and let*

$$u^2 = \mathbb{D}_1 e^1 + \tilde{\mathbb{D}}_2^* b^3 \quad (3.4)$$

denote its discrete Hodge decomposition with respect to the inner product induced by $\tilde{\mathbb{M}}_2$. The pair (a^2, e^1) , where $a^2 = \tilde{\mathbb{D}}_2^ b^3$, solves the linear system*

$$\begin{bmatrix} \mathbb{M}_2 + \mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \tilde{\mathbb{M}}_2 \mathbb{D}_1 \mathbb{M}_1^{-1} \mathbb{D}_1^T \tilde{\mathbb{M}}_2 & \mathbb{M}_2 \mathbb{D}_1 \\ \mathbb{D}_1^T \mathbb{M}_2 & \mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 \end{bmatrix} \begin{bmatrix} a^2 \\ e^1 \end{bmatrix} = \begin{bmatrix} \mathbb{M}_2 f^2 \\ \mathbb{D}_1^T \mathbb{M}_2 f^2 \end{bmatrix}. \quad (3.5)$$

Proof. Denoting $a^2 = \tilde{\mathbb{D}}_2^* b^3$, and applying the decomposition (3.4) to the weak form (3.2) gives

$$(\mathbb{D}_1 e^1 + a^2, \hat{u}^2)_{C_{\Gamma}^2} + (\mathbb{D}_2 a^2, \mathbb{D}_2 \hat{u}^2)_{C^3} = (f^2, \hat{u}^2)_{C_{\Gamma}^2}, \quad \forall \hat{u}^2 \in C_{\Gamma}^2.$$

In the above equality, we used the fact that $\mathbb{D}_2 \mathbb{D}_1 \equiv 0$. We note that the assumed Hodge decomposition implies that $\tilde{\mathbb{D}}_1^* a^2 = 0$ (since $\tilde{\mathbb{D}}_1^* \tilde{\mathbb{D}}_2^* = 0$), thus

$$(\tilde{\mathbb{D}}_1^* a^2, \tilde{\mathbb{D}}_1^* \hat{u}^2)_{C_{\Gamma}^1} \equiv 0, \quad \forall \hat{u}^2 \in C_{\Gamma}^2.$$

As a result, this term can be added to the last equation to obtain:

$$(\mathbb{D}_1 e^1 + a^2, \hat{u}^2)_{C_{\Gamma}^2} + (\mathbb{D}_2 a^2, \mathbb{D}_2 \hat{u}^2)_{C^3} + (\tilde{\mathbb{D}}_1^* a^2, \tilde{\mathbb{D}}_1^* \hat{u}^2)_{C_{\Gamma}^1} = (f^2, \hat{u}^2)_{C_{\Gamma}^2}, \quad \forall \hat{u}^2 \in C_{\Gamma}^2.$$

It is easy to see that the above weak form is equivalent to the following linear system:

$$\mathbb{M}_2 a^2 + \left(\mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \tilde{\mathbb{M}}_2 \mathbb{D}_1 \mathbb{M}_1^{-1} \mathbb{D}_1^T \tilde{\mathbb{M}}_2 \right) + \mathbb{M}_2 \mathbb{D}_1 e^1 = \mathbb{M}_2 f^2$$

which is the first equation in (3.5).

Applying the decomposition (3.4) to (3.1), and then multiplying by \mathbb{D}_1^* on both sides gives

$$\mathbb{D}_1^* a^2 + \mathbb{D}_1^* \mathbb{D}_1 e^1 = \mathbb{D}_1^* f^2.$$

Noticing that by definition $\mathbb{D}_1^* = \mathbb{M}_1^{-1} \mathbb{D}_1^T \mathbb{M}_2$, the second set of equations in the block system follows by multiplying \mathbb{M}_1 on both sides. This completes the proof. \square

Here, we should notice that the (2,2) block $\mathbb{D}_1^* \mathbb{D}_1$ is singular. A further decomposition [5, Corollary 3.2] of

$$e^1 = \mathbb{D}_0 e^0 + \tilde{\mathbb{D}}_1^* b^2 := \mathbb{D}_0 e^0 + a^1$$

yields the following block system

$$\begin{bmatrix} A_{11} & \mathbb{M}_2 \mathbb{D}_1 \\ \mathbb{D}_1^T \mathbb{M}_2 & A_{22} \end{bmatrix} \begin{bmatrix} a^2 \\ a^1 \end{bmatrix} = \begin{bmatrix} \mathbb{M}_2 f^2 \\ \mathbb{D}_1^T \mathbb{M}_2 f^2 \end{bmatrix} \quad (3.6)$$

where $A_{11} = \mathbb{M}_2 + \mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \tilde{\mathbb{M}}_2 \mathbb{D}_1 \mathbb{M}_1^{-1} \mathbb{D}_1^T \tilde{\mathbb{M}}_2$ and $A_{22} = \mathbb{D}_1^T \mathbb{M}_2 \mathbb{D}_1 + \tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1} \mathbb{D}_0 \tilde{\mathbb{M}}_1$. In the above formulation, we used the fact that $\mathbb{D}_1 \mathbb{D}_0 = 0$ and $\tilde{\mathbb{D}}_0^* \tilde{\mathbb{D}}_1^* = 0$.

REMARK 3.2. *The reformulation (3.6) seems more complicated than the original equation (3.1) that we are actually solving. The idea here is try to use the diagonal blocks A_{11} and A_{22} as preconditioner of (3.1), which is the main focus of the next section.*

It is interesting to notice that during this reformulation procedure, the gauge term in the A_{11} and A_{22} blocks seems to be indispensable. As was point out in [5], these terms play an important roles in avoiding the large null-space caused by $\nabla \nabla \cdot$ operator and $\nabla \times \nabla \times$ operator respectively. While to form $\tilde{\mathbb{M}}_2 \mathbb{D}_1 \mathbb{M}_1^{-1} \mathbb{D}_1^T \tilde{\mathbb{M}}_2$ and $\tilde{\mathbb{M}}_1 \mathbb{D}_0 \mathbb{M}_0^{-1} \mathbb{D}_0 \tilde{\mathbb{M}}_1$ requires the inversion of \mathbb{M}_1 and \mathbb{M}_0 . Even if we can use mass lumping to simplify the computation, it makes the system more complicated and ruins the sparse patten of the original system. The interesting fact is that according to the numerical tests (see Section 5 for more details), it is not so clear now if these gauge terms are necessary or not. We need a more rigorous investigation of the roles of these gauge terms for more complex problems.

4. Multigrid solvers. Now we are in position to combine the reformulation and preconditioning to develop a linear solver for the compatible discretization (3.1) of the $H(\text{div})$ equation (1.1). Similar to the algorithm in [5], we focus on developing the AMG block preconditioners.

The approach considered in this paper focuses on developing AMG methods for the (1,1) and (2,2) blocks in (3.6) separately. Note that these diagonal blocks are Laplace-like. Once constructed, these AMG solvers are combined as a Jacobi-like preconditioner to precondition (1.1).

We propose an AMG technique for the whole 2×2 system which employs a Hiptmair smoother (see for example [11]) at the finest level, but allows subsequent levels and transfers of the (1,1) and (2,2) blocks to be handled with the standard AMG method. To do this, the face element of the (1,1) block and the edge element version of the (2,2) block must be converted to a more standard nodal form on the coarse mesh. This is accomplished by two special prolongators that not only transfers solutions from a coarse to a fine solution but also transfers solutions from a nodal to a face or edge representation, respectively. The net effect of these special prolongators is that the corresponding Galerkin projection of the (1,1) and (2,2) block will, in fact, yield a coarse operator resembling a vector nodal Laplacian which is amenable to any standard AMG method for further coarsening.

4.1. The specialized prolongators. As discussed earlier, in order to use the standard AMG solvers for the (1,1) and (2,2) block, we must convert the face element (for the (1,1)-block) and the edge element (for the (2,2)-block) into the standard nodal form. To do this, we define specialized prolongators P_{11} and P_{22} to transfer solutions from a nodal to a face and edge representation respectively. Instead of introducing the near null-space to define the prolongators as was done in [5], here we make use of the interpolation Π_h^{div} and Π_h^{curl} (see Figure 2.1) as in [2] and [12].

There are many ways to obtain aggregates corresponding to nodes, see [5] for more details. In this paper, for simplicity we use perfect aggregation. By “perfect”, we mean that the aggregates are formed manually. Note that we only need to form these aggregates on the finest level. Once the aggregates are formed, Π_h^{div} and Π_h^{curl} must also be computed. The detailed construction of the special prolongators for the (1,1) and (2,2) block is given in Algorithm 1. Notice that the net effect of P_{11} is to

interpolate coarse nodal quantities to fine face-oriented quantities, and the effect of P_{22} is to interpolate coarse nodal quantities to fine edge-oriented quantities.

Algorithm 1: $[P_{11}, P_{22}] = \text{Coarse_Node_Prolongators}()$

- 1 $\{\mathcal{A}_i\} \leftarrow \text{Aggregate manually};$
- 2 For each fine node n_i and each aggregate \mathcal{A}_j define

$$(P_{nf})_{i,j} = \begin{cases} 1, & \text{if } n_i \in \mathcal{A}_j \\ 0, & \text{otherwise} \end{cases}.$$

- 3 $P_{11} = \Pi_h^{\text{div}} P_{nf};$
 - 4 $P_{22} = \Pi_h^{\text{curl}} P_{nf};$
-

The Galerkin coarse discretizations are given by

$$A_{11}^H = P_{11}^T A_{11} P_{11}, \quad A_{22}^H = P_{22}^T A_{22} P_{22}$$

where A_{11} and A_{22} are the (1,1) and (2,2) block of (3.6), A_{11}^H and A_{22}^H refer to their projections on a coarse mesh, respectively.

4.2. Relaxation. As before, we consider the following hybrid scheme. Suppose that the conjugate gradient iteration is actually applied to (3.1) and that (3.6) is *only used within the preconditioner*. To do this, it is necessary to convert residuals of (3.1) to right hand sides of (3.6) within the preconditioner. This is done by applying $[\mathbb{I} \quad \mathbb{D}_1]^T$ to the residual. Approximate solutions to (3.6) are then converted back to a form suitable for (3.1) via $\mathbb{D}_1 a^1 + a^2$.

Algorithm 2 illustrates such a smoother proposed by Hiptmair that combines standard smoothing of the original equations with standard smoothing of the equations projected to the null-space [11].

Algorithm 2: $\tilde{u} = \text{FineRelaxation}(A, \mathbb{D}_1, \tilde{u}, b)$

- 1 $\tilde{u} \leftarrow \text{StandardRelaxation}(A, \tilde{u}, b);$
 - 2 $c \leftarrow \text{StandardRelaxation}(\mathbb{D}_1^T A \mathbb{D}_1, 0, \mathbb{D}_1^T (b - A\tilde{u}));$
 - 3 $\tilde{u} \leftarrow \tilde{u} + \mathbb{D}_1 c;$
 - 4 $\tilde{u} \leftarrow \text{StandardRelaxation}(A, \tilde{u}, b);$
-

The key is that the error is smooth after this initial relaxation. Since the error is smooth, fine grid relaxation may be omitted from the AMG V-cycles in $\text{Solve}()$, as (3.1) and (1.1) are equivalent.

It is important to realize that this special smoother is only needed on the finest level. A standard smoother can be used on coarse levels within the AMG procedures for the (1,1) and (2,2) blocks. Finally, an additive version of the Hiptmair smoother may also be considered for $\text{FineRelaxation}()$.

4.3. AMG algorithm preconditioner. We now give the entire AMG-based preconditioner for the block Jacobi version in Algorithm 3. $\text{PreFineRelaxation}()$ is identical to Algorithm 2 except step one is omitted. This also avoids the residual calculation in step two as the initial guess to a preconditioner is always zero. $\text{PostFineRelaxation}()$ is identical to Algorithm 2 except step four is omitted to keep

the preconditioner symmetric when `StandardRelaxation()` employs a symmetric algorithm. Of course, residual calculations can also be avoided using additive forms of this smoother.

The algorithm essentially involves two AMG solves for nodal vector Laplacians: A_{11}^H corresponding to the (1,1) block and A_{22}^H corresponding to the (2,2) block. In addition, some relaxation must be performed on the original fine mesh system. Specifically, there are three major components of the preconditioner.

- (1) Hiptmair smoother for $H(\text{div})$ (see also Hiptmair [11]).
- (2) AMG for $P_{11}^T A_{11} P_{11}$ within the (1,1)-block.
- (3) AMG for $P_{22}^T A_{22} P_{22}$ within the (2,2)-block.

The detailed algorithm is listed as follows:

Algorithm 3: $\tilde{u} = \text{Block Preconditioner}(r)$

% Setup Phase

Form $A_{11}^H \leftarrow P_{11}^T A_{11} P_{11}$ efficiently;
 Standard_AMG_Setup(A_{11}^H);
 Form $A_{22}^H \leftarrow P_{22}^T A_{22} P_{22}$ efficiently;
 Standard_AMG_Setup(A_{22}^H);

;

% Solve Phase

$\tilde{u} \leftarrow \text{PreFineRelaxation}(\mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \mathbb{M}_2, \mathbb{D}_1, 0, r)$;
 $\tilde{r} \leftarrow r - (\mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \mathbb{M}_2) \tilde{u}$;
% Perform V-cycles on A_{11}^H and A_{22}^H
 $a \leftarrow \text{Standard_AMG_Vcycle}(A_{11}^H, 0, P_{11}^T \tilde{r})$;
 $p \leftarrow \text{Standard_AMG_Vcycle}(A_{22}^H, 0, P_{22}^T \mathbb{D}_1^T \tilde{r})$;
 $\tilde{u} \leftarrow \tilde{u} + P_{11} a + \mathbb{D}_1 P_{22} p$;
 $\tilde{u} \leftarrow \text{PostFineRelaxation}(\mathbb{D}_2^T \mathbb{M}_3 \mathbb{D}_2 + \mathbb{M}_2, \mathbb{D}_1, \tilde{u}, r)$;

5. Numerical results. All the numerical experiments are conducted in a three-dimensional unit cube domain $\Omega = \{(x, y, z) \in \mathbb{R}^3 : 0 \leq x, y, z \leq 1\}$ with homogeneous Neumann boundary condition. The domain is meshed by uniform cubes, and each cube is divided into 6 tetrahedra.

The proposed solver was implemented using CG in MATLAB. The first level and the first grid transfer of Algorithm 3 is also implemented in MATLAB. ML's smoothed aggregation solver is used for A_{11}^H and A_{22}^H , through the `mlmex` MATLAB interface [9]. A single V-cycle of AMG is used for both the (1,1) and (2,2) block, using the efficient variant of Algorithm 2 (smoother). Unless otherwise stated, we use two steps of symmetric Gauss-Seidel sub-smoothing on both faces and edges. For all experiments the CG tolerance is 1×10^{-10} .

5.1. Constant coefficients. As the first experiment, we consider the constant coefficients case. We assume that $\lambda = \mu = 1$ in Ω . Table 5.1 reports the number of iterations with different meshsize. We note that the number of iterations are almost identical whether we include the gauge terms in the (1,1), and (2,2)-block or not. By this reason, we will omit the gauge term in the following numerical experiments.

TABLE 5.1

Number of iterations for CG-accelerated AMG on the 3D tetrahedral mesh problem with constant coefficients, using Algorithm 3. The size of the problem and the number of SGS smoothing steps are varied.

Grid		9^3	12^3	15^3	18^3	21^3	24^3	27^3
2 SGS Steps	gauge	12	12	13	13	13	13	13
	No gauge	11	13	13	14	14	14	15
3 SGS Steps	gauge	10	11	11	12	12	12	12
	No gauge	9	10	11	12	12	13	13
4 SGS Steps	gauge	9	10	10	11	11	11	11
	No gauge	8	10	10	10	11	11	11

TABLE 5.2

Number of iterations for CG-accelerated AMG on the 3D tetrahedral mesh problem with jump coefficients, using Algorithm 3. μ_0 varies inside $[1/3, 2/3]^3$, and 1 elsewhere, and $\lambda \equiv 1$.

Grid	10^{-8}	10^{-7}	10^{-6}	10^{-5}	μ_0^{-1} 10^{-4}	10^3	10^{-2}	10^{-1}	1
9^3	11	11	11	11	11	11	11	11	11
18^3	15	15	15	15	16	16	15	15	14
27^3	16	16	19	18	18	18	19	17	15

5.2. Variable μ . We experiment with jumps in μ by considering two regions with constant values of μ . Specifically, define

$$\Omega_0 = \left\{ (x, y, z) : \frac{1}{3} \leq x, y, z \leq \frac{2}{3} \right\}, \quad \Omega_1 = \Omega \setminus \Omega_0;$$

let $\mu \equiv 1$ in Ω_1 and choose $\mu = \mu_0$ to be a constant inside Ω_0 . λ is fixed to be 1 throughout the whole domain Ω . Table 5.2 reports the number of iterations on different meshsize. Note that the number of iterations are quite robust with respect to the variation of the coefficient μ .

5.3. Variable λ . We now consider the jump on λ . Same as before, we choose $\lambda = \lambda_0$ to be a constant which varies from 10^{-4} to 10^4 inside the domain Ω_0 , and $\lambda = 1$ elsewhere. This time, we fix μ to be 1 in the whole domain Ω . Table 5.3 reports the number of iterations on different meshsize. Again, the number of iterations remains fairly constant.

6. Conclusions. In this paper, we proposed an AMG based preconditioner for the $H(\text{div})$ equation. We reformulated the equation by using the compatible gauge approaches, and formed a 2×2 system which is equivalent to the original discrete linear equations. Then we combined the AMG solvers for the (1,1) and (2,2) blocks of this system in certain way, and used it as the preconditioner of the original linear system. We also presented some numerical experiments to show the robustness of this algorithm. These experiments showed that the algorithm is very robust even with the presence of large jump coefficients.

TABLE 5.3

Number of iterations for CG-accelerated AMG on the 3D tetrahedral mesh problem with jump coefficients, using Algorithm 3. λ_0 varies inside $[1/3, 2/3]^3$, and 1 elsewhere, and $\mu \equiv 1$.

Grid	λ_0								
	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^1	10^2	10^3	10^4
9^3	17	16	14	12	11	11	11	11	9
18^3	21	20	18	16	14	14	14	12	12
27^3	22	21	21	17	15	15	14	13	13

- [1] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Preconditioning in $H(\text{div})$ and applications*, MATHEMATICS OF COMPUTATION, 66 (1997), pp. 957–984.
- [2] R. BECK, *Algebraic multigrid by component splitting for edge elements on simplicial triangulations*, Tech. Report SC 99-40, Zuse Institute Berlin, December 1999.
- [3] P. BOCHEV AND J. HYMAN, *Principles of mimetic discretizations of differential operators*, in Compatible Spatial Discretizations, D. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, eds., Springer-Verlag, 2006.
- [4] P. BOCHEV AND A. ROBINSON, *Matching algorithms with physics: exact sequences of finite element spaces*, in Preservation of stability under discretization, D. Estep and S. Tavener, eds., Philadelphia, 2001, SIAM, pp. 145–165.
- [5] P. B. BOCHEV, J. J. HU, C. M. SIEFERT, AND R. S. TUMINARO, *An algebraic multigrid approach based on a compatible gauge reformulation of Maxwell’s equations*, Tech. Report SAND2007-1633J, Sandia National Laboratory, 2007.
- [6] F. BREZZI AND M. FORTIN, *Mixed and Hybrid Finite Element Methods*, vol. 15 of Springer series in computational mathematics, Springer-Verlag, 1991.
- [7] S. CAIRNS, *Introductory topology*, Ronald Press Co., New York, 1961.
- [8] A. DEZIN, *Multidimensional analysis and discrete models*, CRC Press, Boca Raton, 1995.
- [9] M. GEE, C. SIEFERT, J. HU, R. TUMINARO, AND M. SALA, *ML 5.0 smoothed aggregation user’s guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, 2006.
- [10] V. GIRAUT AND P. A. RAVIART, *Finite element methods for Navier-Stokes equations*, Springer-Verlag, Berlin, 1986. Theory and algorithms.
- [11] R. HIPTMAIR, *Multigrid method for $H(\text{div})$ in three dimensions*, Electron. Trans. Numer. Anal., 6 (1997), pp. 133–152. Special issue on multilevel methods (Copper Mountain, CO, 1997).
- [12] R. HIPTMAIR AND J. XU, *Nodal auxiliary space preconditioning in $H(\text{curl})$ and $H(\text{div})$ spaces*, tech. report, 2006.
- [13] J. HYMAN AND M. SHASHKOV, *Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids*, Appl. Num. Math., 25 (1997), pp. 413–442.
- [14] ———, *Natural discretizations for the divergence, gradient and curl on logically rectangular grids*, Comput. Math. Appl., 33 (1997), pp. 88–104.
- [15] R. NICOLAIDES, *Direct discretization of planar div-curl problems*, SIAM J. Numer. Anal., 29 (1992), pp. 32–56.
- [16] M. SHASHKOV, *Conservative finite difference methods on general grids*, CRC Press, Boca Raton, FL, 1996.
- [17] J. VAN WELIJ, *Calculation of eddy currents in terms of H on hexahedra*, IEEE Transactions on Magnetics, 21 (1985), pp. 2239–2241.
- [18] P. S. VASSILEVSKI AND J. WANG, *Multilevel iterative methods for mixed finite element discretizations of elliptic problems*, Numerische Mathematik, 63 (1992), pp. 503–520.
- [19] B. I. WOHLMUTH, A. TOSELLI, AND O. B. WIDLUND, *An iterative substructuring method for Raviart–Thomas vector fields in three dimensions*, SIAM Journal on Numerical Analysis, 37 (2000), pp. 1657–1676.